



(19)

(11) Publication number: 2000003268 A

Generated Document.

PATENT ABSTRACTS OF JAPAN

(21) Application number: 10163987

(51) Intl. Cl.: G06F 9/06 G06F 9/45 G06F 12/06

(22) Application date: 11.06.98

(30) Priority:

(43) Date of application
publication: 07.01.00(84) Designated
contracting states:

(71) Applicant: MITSUBISHI ELECTRIC CORP

(72) Inventor: HAGIWARA MASATO

(74) Representative:

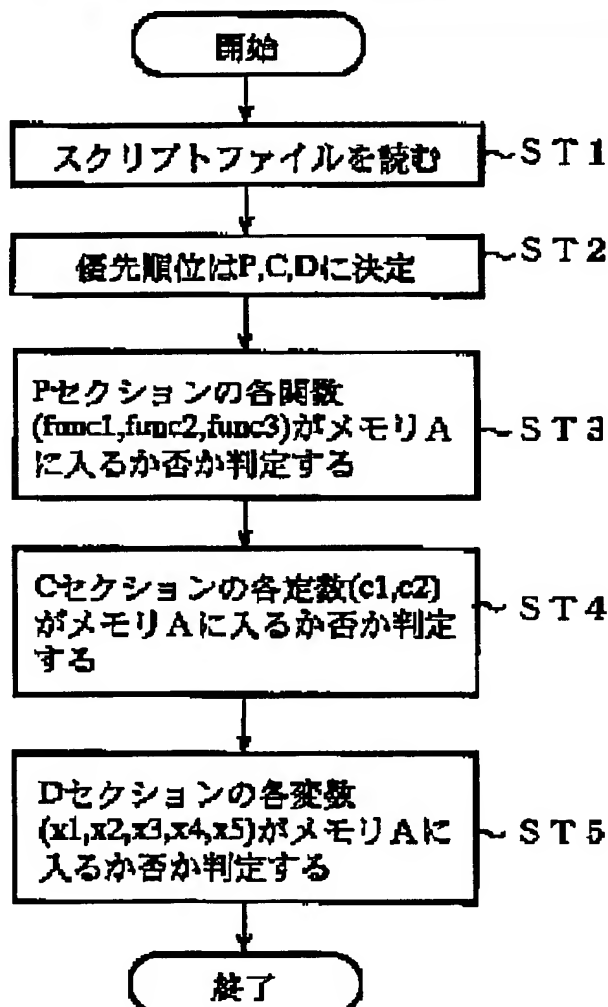
(54) METHOD FOR DECIDING
OPTIMIZATION OF MEMORY
MAP

(57) Abstract:

PROBLEM TO BE SOLVED: To obtain a method for deciding memory map optimization for quickening the executing speed of a program on a system on which plural kinds of memories whose access speed is different are loaded, and for easily constructing an optimized memory map.

SOLUTION: This method comprises priority order deciding steps (ST1 and ST2) for deciding a priority order for programs to be arranged in the order of a memory whose access speed is fast and memory map optimizing steps (ST3-ST5) for deciding a memory and the area sufficient for arranging each program in the order of the memory whose access speed is fast based on the priority order decided in the priority order deciding steps by a linker, and for deciding the memory map of the program operating on the system on which plural kinds of memories whose access speed is different.

COPYRIGHT: (C)2000,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-3268

(P2000-3268A)

(43) 公開日 平成12年1月7日 (2000.1.7)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 9/06	4 1 0	G 0 6 F 9/06	4 1 0 E 5 B 0 6 0
9/45		12/06	5 2 2 D 5 B 0 7 6
12/06	5 2 2	9/44	3 2 2 H 5 B 0 8 1

審査請求 未請求 請求項の数 9 O L (全 11 頁)

(21) 出願番号 特願平10-163987

(22) 出願日 平成10年6月11日 (1998.6.11)

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 萩原 正人

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(74) 代理人 100066474

弁理士 田澤 博昭 (外1名)

Fターム(参考) 5B060 MM03

5B076 AB02

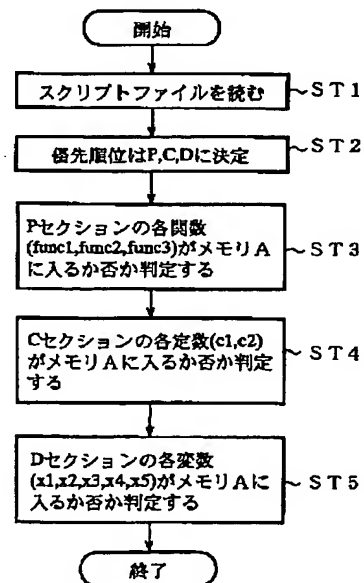
5B081 CC27

(54) 【発明の名称】 メモリマップ最適化決定方法

(57) 【要約】

【課題】 アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを容易に構築できるメモリマップ最適化決定方法を得ることである。

【解決手段】 アクセス速度の早いメモリの順で配置すべきプログラムに対する優先順位を決定する優先順位決定ステップと、該優先順位決定ステップで決定した優先順位をもとに、前記アクセス速度の早いメモリから順に前記各プログラムを配置するに足りるメモリおよびその領域をリンカーが決定し、アクセス速度の異なる複数種類のメモリを搭載したシステム上で動作するプログラムのメモリマップを決定するメモリマップ最適化ステップとを備える。



ステップST1, ステップST2: (優先順位決定ステップ)
ステップST3~ステップST5: (メモリマップ最適化ステップ)

【特許請求の範囲】

【請求項1】 アクセス速度の早いメモリの順で配置すべきプログラムに対する優先順位を決定する優先順位決定ステップと、

該優先順位決定ステップで決定した優先順位をもとに、前記アクセス速度の早いメモリから順に前記各プログラムを配置するに足りるメモリおよびその領域をリンカーが決定し、アクセス速度の異なる複数種類のメモリを搭載したシステム上で動作するプログラムのメモリマップを決定するメモリマップ最適化ステップとを備えたメモリマップ最適化決定方法。

【請求項2】 優先順位決定ステップでは、スクリプトファイルに記述された各プログラムに対する優先順位をもとに、前記各プログラムの優先順位を決定する請求項1記載のメモリマップ最適化決定方法。

【請求項3】 優先順位決定ステップでは、リンカー起動時にコマンドラインからオプションとして指定された優先順位をもとに、各プログラムの優先順位を決定する請求項1記載のメモリマップ最適化決定方法。

【請求項4】 優先順位決定ステップでは、各オブジェクトファイルにおいて他のオブジェクトファイルと共通的に関数毎、変数毎、定数毎に分類されたセクション毎で優先順位を決定することを特徴とする請求項1から請求項3のうちのいずれか1項記載のメモリマップ最適化決定方法。

【請求項5】 優先順位決定ステップでは、各オブジェクトファイルの関数毎、変数毎、定数毎に優先順位を決定することを特徴とする請求項1から請求項3のうちのいずれか1項記載のメモリマップ最適化決定方法。

【請求項6】 優先順位決定ステップでは、オブジェクトファイル毎に優先順位を決定することを特徴とする請求項1から請求項3のうちのいずれか1項記載のメモリマップ最適化決定方法。

【請求項7】 優先順位決定ステップでは、他のオブジェクトファイルとの間で共通的に各オブジェクトファイルにおいて関数毎、変数毎、定数毎に分類されたセクション毎と、各オブジェクトファイルの関数毎、変数毎、定数毎と、オブジェクトファイル毎との内のいずれかの組み合わせで優先順位を決定することを特徴とする請求項1から請求項3のうちのいずれか1項記載のメモリマップ最適化決定方法。

【請求項8】 優先順位決定ステップでは、リンカーが全てのオブジェクトファイルに対して各関数、変数、定数などにアクセスしている箇所をカウントし、そのカウント値をもとに優先順位を自動的に決定することを特徴とする請求項1記載のメモリマップ最適化決定方法。

【請求項9】 優先順位決定ステップでは、

プログラムを実行させてプロファイルした結果をもとに優先順位を決定することを特徴とする請求項1記載のメモリマップ最適化決定方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、アクセス速度の異なる複数種類のメモリを搭載したシステム上で高速動作するプログラムを開発する際に用いて好適な、プログラムの高速動作化に最適なメモリマップを容易に決定できるメモリマップ最適化決定方法に関するものである。

【0002】

【従来の技術】複数種類のメモリを搭載したシステムでは各メモリに対するアクセス速度は大きく異なる場合が多い。従ってプログラムのどの部分をどのメモリに配置するかによってプログラムの実行速度は大きく異なってくる。そのため、プログラムの実行速度を高速化するメモリマップを決定するにはアクセス頻度の高い部分をなるべく速いメモリに配置するなどの考慮が必要となる。このため、リンカーにコマンドラインオプションやスクリプトファイルなどで各セクションの先頭アドレスを指定することによってメモリ上の配置を決定していた。

【0003】

【発明が解決しようとする課題】従来のメモリマップ最適化決定方法は以上のように構成されているので、各セクションのサイズと各メモリの容量が問題になってくる。つまり優先度の高いセクションが速いメモリに収まりきれぬか否かなどを考慮する必要があるため、各セクションのサイズをあらかじめ調べておいてからサイズとアクセス頻度の両面を考慮しつつ最適なメモリマップを決定する必要がある課題があった。

【0004】また、プログラムを少し修正しただけでも各セクションのサイズが変わってしまうため、そのたびにメモリマップを考え直す必要があり、この作業には手間がかかる課題があった。

【0005】この発明は上記のような課題を解決するためになされたもので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを容易に構築できるメモリマップ最適化決定方法を得ることを目的とする。

【0006】

【課題を解決するための手段】この発明に係るメモリマップ最適化決定方法は、アクセス速度の早いメモリの順で配置すべきプログラムに対する優先順位を決定する優先順位決定ステップと、該優先順位決定ステップで決定した優先順位をもとに、前記アクセス速度の早いメモリから順に前記各プログラムを配置するに足りるメモリおよびその領域をリンカーが決定し、アクセス速度の異なる複数種類のメモリを搭載したシステム上で動作するプログラムのメモリマップを決定するメモリマップ最適化

ステップとを備えるようにしたものである。

【0007】この発明に係るメモリマップ最適化決定方法は、スクリプトファイルに記述された各プログラムに対する優先順位をもとに、前記各プログラムの優先順位を決定する優先順位決定ステップを備えるようにしたものである。

【0008】この発明に係るメモリマップ最適化決定方法は、リンカー起動時にコマンドラインからオプションとして指定された優先順位をもとに、各プログラムの優先順位を決定する優先順位決定ステップを備えるようにしたものである。

【0009】この発明に係るメモリマップ最適化決定方法は、各オブジェクトファイルにおいて他のオブジェクトファイルと共通的に関数毎、変数毎、定数毎に分類されたセクション毎で優先順位を決定する優先順位決定ステップを備えるようにしたものである。

【0010】この発明に係るメモリマップ最適化決定方法は、各オブジェクトファイルの関数毎、変数毎、定数毎に優先順位を決定する優先順位決定ステップを備えるようにしたものである。

【0011】この発明に係るメモリマップ最適化決定方法は、オブジェクトファイル毎に優先順位を決定する優先順位決定ステップを備えるようにしたものである。

【0012】この発明に係るメモリマップ最適化決定方法は、他のオブジェクトファイルとの間で共通的に各オブジェクトファイルにおいて関数毎、変数毎、定数毎に分類されたセクション毎と、各オブジェクトファイルの関数毎、変数毎、定数毎と、オブジェクトファイル毎との内のいずれかの組み合わせで優先順位を決定する優先順位決定ステップを備えるようにしたものである。

【0013】この発明に係るメモリマップ最適化決定方法は、リンカーが全てのオブジェクトファイルに対して各関数、変数、定数などにアクセスしている箇所をカウントし、そのカウント値をもとに優先順位を自動的に決定する優先順位決定ステップを備えるようにしたものである。

【0014】この発明に係るメモリマップ最適化決定方法は、プログラムを実行させてプロファイルした結果をもとに優先順位を決定する優先順位決定ステップを備えるようにしたものである。

【0015】

【発明の実施の形態】以下、この発明の実施の一形態を説明する。

実施の形態1. 図1は、この実施の形態1のメモリマップ最適化決定方法におけるサンプルプログラムの構成の一例を示す説明図である。図1に示す例では、プログラムは3つのオブジェクトファイルF1、F2、F3から構成されている。オブジェクトファイルF1ではサイズが4である関数f u n c 1、サイズが1である変数x1、サイズが2である定数c1が定義されている。ま

た、オブジェクトファイルF2ではサイズが7の関数f u n c 2、サイズが1である変数x2、サイズが1である定数c2が定義されている。また、オブジェクトファイルF3ではサイズが3である関数f u n c 3、サイズが2である変数x3、サイズが1である変数x4、サイズが1である変数x5が定義されている。なお、各オブジェクトファイルにおいては、関数がPセクション、変数がDセクション、定数はCセクションとしてそれぞれ分類されている。

【0016】この実施の形態1では、これらのオブジェクトファイルF1、F2、F3をリンクして、速いメモリA(サイズ10)と遅いメモリB(サイズ20)を搭載したシステム上にマップする場合を例に説明を行う。

【0017】図2は、この実施の形態1のメモリマップ最適化決定方法の構成を示すフローチャートであり、以下、このフローチャートに基づいて動作を説明する。図2に示すフローチャートは、速いメモリに配置する優先順位を、セクション単位でスクリプトファイル(ファイル)に記述することによって指定し、図1のプログラムの最適化されたメモリマップを決定する処理動作を示す。なお、この場合、前記優先順位はPセクション、Cセクション、Dセクションの順で指定するものとする。

【0018】まず、前記スクリプトファイルにセクション単位で記述された速いメモリに配置する前記優先順位を読み込む(ステップST1、優先順位決定ステップ)。次に、前記読み込んだ優先順位から各セクション間の優先順位を決定する(ステップST2、優先順位決定ステップ)。この場合、前記優先順位はPセクション、Cセクション、Dセクションの順である。

【0019】このため、最も優先順位の高いPセクションに含まれる各オブジェクトファイルF1、F2、F3の各関数について速いメモリAに収まるかを調べる。この結果、オブジェクトファイルF1のf u n c 1はサイズ4でありサイズ10のメモリAに収まるので、メモリAに配置する。次に、オブジェクトファイルF2のf u n c 2はサイズ7でありメモリAの残りサイズ6に収まらないのでメモリBに配置する。さらにオブジェクトファイルF3のf u n c 3はサイズ3でありメモリAの残りサイズ6に収まるのでメモリAに配置する(ステップST3、メモリマップ最適化ステップ)。

【0020】次に、優先順位の高いCセクションに含まれる各オブジェクトファイルF1、F2、F3の各定数について速いメモリAに収まるかを調べる。この結果、オブジェクトファイルF1の定数c1はサイズ2でありメモリAの残りサイズ3に収まるのでメモリAに配置する。次にオブジェクトファイルF2の定数c2はサイズ1でありメモリAの残りサイズ1に収まるのでメモリAに配置する(ステップST4、メモリマップ最適化ステップ)。

【0021】この時点で、メモリAの領域には配置割当

ての可能な領域がなくなるため、各オブジェクトファイルF1、F2、F3の残りのプログラムは全てメモリBに配置する。この結果、メモリBの残りの領域には、オブジェクトファイルF3の変数x3、x4、x5が配置される(ステップST5、メモリマップ最適化ステップ)。

【0022】図3は、このようにして、各オブジェクトファイルの各セクション単位で決定された優先順位をもとに、プログラムの実行速度の高速化が図れるように早いメモリAと遅いメモリBに対し最適化されて構築されたメモリマップを示す。

【0023】以上のように、この実施の形態1によれば、オブジェクトファイルの各セクション単位毎に決定された優先順位で、高速動作が要求される前記セクション単位のプログラムをアクセスの早いメモリA、遅いメモリBの順で容易に割り当て配置することができ、プログラムの実行速度の高速化を図れる最適化されたメモリマップの構築作業にかかる負荷を軽減できるメモリマップ最適化決定方法が得られる効果がある。

【0024】実施の形態2。図4は、この実施の形態2のメモリマップ最適化決定方法の構成を示すフローチャートである。この実施の形態2では、速いメモリAに配置する優先順位を、変数やモジュール単位でスクリプトファイルに記述することによって指定する。なお、この実施の形態2では前記優先順位は関数func1、変数x3、関数func3、定数c1、変数x1の順であり、その他の優先順位は変数x1よりも低いレベルであるものとする。

【0025】以下、図4のフローチャートに従って説明すると、まず前記スクリプトファイルに変数やモジュール単位で記述されている前記優先順位を読み込み(ステップST11、優先順位決定ステップ)、優先順位が関数func1、変数x3、関数func3、定数c1、変数x1の順であることを決定する(ステップST12、優先順位決定ステップ)。

【0026】次に、関数func1がメモリAに入るか否かを判定する。関数func1はサイズ4でありサイズ10のメモリAに収まるので、メモリAに配置する(ステップST13、メモリマップ最適化ステップ)。

【0027】次に、変数x3がメモリAに入るか否かを判定する。変数x3はサイズ2なのでメモリAの残りサイズ6に収まるので、メモリAに配置する(ステップST14、メモリマップ最適化ステップ)。

【0028】次に、関数func3がメモリAに入るか否かを判定する。関数func3はサイズ3でありメモリAの残りサイズ4に収まるので、メモリAに配置する(ステップST15、メモリマップ最適化ステップ)。

【0029】次に、定数c1がメモリAに入るか否かを判定する。定数c1はサイズ2なのでメモリAの残りサイズ1に収まらないのでメモリBに配置する(ステップ

ST16、メモリマップ最適化ステップ)。

【0030】次に、変数x1がメモリAに入るか否かを判定する。変数x1はサイズ1でありメモリAの残りサイズ1に収まるので、メモリAに配置する(ステップST17、メモリマップ最適化ステップ)。

【0031】この時点でメモリAには配置割当て可能な領域がなくなるため残りは全てメモリBに配置する(ステップST18、メモリマップ最適化ステップ)。

【0032】図5は、このようにして、各オブジェクトファイルの関数func1、変数x3、関数func3、定数c1、変数x1などの順で決定された優先順位をもとにプログラムの実行速度の高速化が図れるように早いメモリAと遅いメモリBに対し最適化されて構築されたメモリマップを示す。

【0033】以上のように、この実施の形態2によれば、変数やモジュール単位毎に決定された優先順位で、高速動作が要求される前記変数やモジュール単位のプログラムをアクセスの早いメモリA、遅いメモリBの順で容易に割り当て配置することが、できプログラムの実行速度の高速化を図れる最適化されたメモリマップの構築作業にかかる負荷を軽減できるメモリマップ最適化決定方法が得られる効果がある。

【0034】実施の形態3。図6は、この実施の形態3のメモリマップ最適化決定方法の構成を示すフローチャートである。この実施の形態3では、速いメモリAに配置する優先順位を、オブジェクトファイル単位でスクリプトファイルに記述することによって指定する。なお、この実施の形態3では前記優先順位はオブジェクトファイルF1、オブジェクトファイルF3、オブジェクトファイルF2の順であるものとする。

【0035】以下、図6のフローチャートに従って説明すると、まず、前記スクリプトファイルにオブジェクトファイル単位で記述されている前記優先順位を読み込み(ステップST21、優先順位決定ステップ)、優先順位がオブジェクトファイルF1、オブジェクトファイルF3、オブジェクトファイルF2の順であることを決定する(ステップST22、優先順位決定ステップ)。

【0036】次に、オブジェクトファイルF1に含まれる関数func1、変数x1、定数c1についてメモリAに収まるか調べる(ステップST23、メモリマップ最適化ステップ)。この場合、関数func1はサイズ4であり、サイズ10のメモリAに収まるので、メモリAに配置する。次に、オブジェクトファイルF1の変数x1はサイズ1であり、メモリAの残りサイズ6に収まるので、メモリAに配置する。さらに、オブジェクトファイルF1の定数c1はサイズ2であり、メモリAの残りサイズ5に収まるので、メモリAに配置する。

【0037】続いて、オブジェクトファイルF3に含まれる関数func3、変数x3、変数x4、変数x5についてメモリAに収まるか調べる(ステップST24、

メモリマップ最適化ステップ)。この場合、関数 `func3` はサイズ3でありメモリAの残りサイズ3に収まるので、メモリAに配置する。この時点で、メモリAは配置割当て可能な領域がなくなるため、残りは全てメモリBに配置する(ステップST25、メモリマップ最適化ステップ)。

【0038】図7は、このようにして各オブジェクトファイル単位で決定された優先順位をもとに早いメモリAと遅いメモリBに対しプログラムの実行速度の高速化が図れるように最適化されて構築されたメモリマップを示す。

【0039】以上のように、この実施の形態3によれば、オブジェクトファイル単位毎に決定された優先順位で、高速動作が要求される前記オブジェクトファイルのプログラムをアクセスの早いメモリA、遅いメモリBの順で容易に割り当て配置することができ、プログラムの実行速度の高速化を図れる最適化されたメモリマップの構築作業にかかる負荷を軽減できるメモリマップ最適化決定方法が得られる効果がある。

【0040】実施の形態4。図8は、この実施の形態4のメモリマップ最適化決定方法の構成を示すフローチャートである。この実施の形態4では、速いメモリAに配置する優先順位を、セクション単位、モジュール単位、オブジェクトファイル単位をおり混ぜてスクリプトファイルに記述することによって指定する。なお、この実施の形態4では前記優先順位は関数 `func1`、オブジェクトファイルF3中のDセクション、Cセクションの順であるものとする。

【0041】以下、図8のフローチャートに従って説明すると、まず、前記スクリプトファイルに関数 `func1`、オブジェクトファイルF3中のDセクション、Cセクションの順で記述されている前記優先順位を読み込み(ステップST31、優先順位決定ステップ)、優先順位が関数 `func1`、オブジェクトファイルF3中のDセクション、Cセクションの順であることを決定する(ステップST32、優先順位決定ステップ)。

【0042】次に、関数 `func1` がメモリAに配置可能かを判定する。関数 `func1` はサイズ4であり、サイズ10のメモリAに収まるので、メモリAに配置する(ステップST33、メモリマップ最適化ステップ)。続いて、オブジェクトファイルF3に含まれるDセクションの変数 `x3`、変数 `x4`、変数 `x5` についてメモリAに収まるか調べる。変数 `x3` はサイズ2でありメモリAの残りサイズ6に収まるので、メモリAに配置する。また、変数 `x4` はサイズ1でありメモリAの残りサイズ4に収まるので、メモリAに配置する。また変数 `x5` はサイズ1なのでメモリAの残りサイズ3に収まるので、メモリAに配置する(ステップST34、メモリマップ最適化ステップ)。

【0043】続いて、Cセクションである定数 `c1`、`c`

2についてメモリAに収まるか判定する(ステップST35、メモリマップ最適化ステップ)。定数 `c1` はサイズ2なのでメモリAの残りサイズ2に収まるのでメモリAに配置する。

【0044】この時点でメモリAは配置割当て可能な領域がなくなるため、残りは全てメモリBに配置する(ステップST36、メモリマップ最適化ステップ)。

【0045】図9は、このようにして関数 `func1`、オブジェクトファイルF3中のDセクション、Cセクションの順で決定された優先順位をもとに、早いメモリAと遅いメモリBに対しプログラムの実行速度の高速化が図れるように最適化されて構築されたメモリマップを示す。

【0046】以上のように、この実施の形態4によれば、セクション単位、モジュール単位、オブジェクトファイル単位などに渡って決定された優先順位で、高速動作が要求される前記セクション単位、モジュール単位、オブジェクトファイル単位などのプログラムをアクセスの早いメモリA、遅いメモリBの順で容易に割り当て配置することができ、プログラムの実行速度の高速化を図れる最適化されたメモリマップの構築作業にかかる負荷を軽減できるメモリマップ最適化決定方法が得られる効果がある。

【0047】実施の形態5。以上説明した各実施の形態では、優先順位の指定をスクリプトファイルに記述する構成であったが、コマンドラインオプションで指定するようにしてもよい。

【0048】実施の形態6。この実施の形態6では、優先順位の指定をスクリプトファイルに記述するのでなく、リンカーが全てのオブジェクトファイルに対して各関数、変数、定数などにアクセスしている箇所をカウントし、そのカウント値をもとに優先順位を自動的に決定する。図10は、この実施の形態6のメモリマップ最適化決定方法の構成を示すフローチャートである。まず、リンカーが全てのオブジェクトファイルに対して各関数、変数、定数などにアクセスしている箇所をカウントし、そのカウント値をもとに優先順位を決定する(ステップST41、優先順位決定ステップ)。この場合、アクセス箇所の検索の結果、関数 `func1`、変数 `x3`、関数 `func3`、定数 `c1`、変数 `x1`、関数 `func2`、変数 `x2`、変数 `x4`、変数 `x5`、定数 `c2` の順に優先順位を自動的に決定したものとする(ステップST42、優先順位決定ステップ)。

【0049】次に、関数 `func1` がメモリAに入るか否かを判定する。関数 `func1` はサイズ4でありサイズ10のメモリAに収まるので、メモリAに配置する(ステップST43、メモリマップ最適化ステップ)。

【0050】次に、変数 `x3` がメモリAに入るか否かを判定する。変数 `x3` はサイズ2なのでメモリAの残りサイズ6に収まるので、メモリAに配置する(ステップS

T44、メモリマップ最適化ステップ)。

【0051】次に、関数 `func3` がメモリAに入るか否かを判定する。関数 `func3` はサイズ3なのでメモリAの残りサイズ4に収まるので、メモリAに配置する(ステップST45、メモリマップ最適化ステップ)。

【0052】次に、定数 `c1` がメモリAに入るか否かを判定する。定数 `c1` はサイズ2なのでメモリAの残りサイズ1に収まらないのでメモリBに配置する(ステップST46、メモリマップ最適化ステップ)。

【0053】次に、変数 `x1` がメモリAに入るか否かを判定する。変数 `x1` はサイズ1なのでメモリAの残りサイズ1に収まるので、メモリAに配置する(ステップST47、メモリマップ最適化ステップ)。

【0054】この時点でメモリAは配置割当て可能な領域がなくなるため、残りは全てメモリBに配置する(ステップST48、メモリマップ最適化ステップ)。

【0055】図11は、このようにしてリンカーにより自動的に決定された優先順位をもとに早いメモリAと遅いメモリBに対しプログラムの実行速度の高速化を図れるように最適化されて構築されたメモリマップを示す。

【0056】以上のように、この実施の形態6によれば、全てのオブジェクトファイルに対して各関数、変数、定数などにアクセスしている箇所のカウント値をもとにリンカーが自動的に決定した優先順位をもとに、高速動作が要求される前記関数、変数、定数などのプログラムをアクセスの早いメモリA、遅いメモリBの順で容易に割り当て配置することができ、プログラムの実行速度の高速化を図れる最適化されたメモリマップの構築作業にかかる負荷を軽減できるメモリマップ最適化決定方法が得られる効果がある。

【0057】実施の形態7。この実施の形態7では、一度プログラムを実行させてプロファイルした結果をもとに優先順位を決定する。この実施の形態7では、プロファイルの結果、関数 `func1`、変数 `x3`、関数 `func3`、定数 `c1`、変数 `x1`、関数 `func2`、変数 `x2`、変数 `x4`、変数 `x5`、定数 `c2` の順で優先順位が決定したものとす。図12は、この実施の形態7のメモリマップ最適化決定方法の構成を示すフローチャートである。なお、図12に示すフローチャートにおいて図10と同一または相当のステップには同一の符号を付して説明を省略する。まず、プログラムを実際に実行し、またはシミュレータなどの使用によりプロファイルした結果をもとに優先順位を決定する(ステップST51、優先順位決定ステップ)。この場合、関数 `func1`、変数 `x3`、関数 `func3`、定数 `c1`、変数 `x1`、関数 `func2`、変数 `x2`、変数 `x4`、変数 `x5`、定数 `c2` の順で優先順位が決定するものとす(ステップST52、優先順位決定ステップ)。次のステップST43以下の各ステップの動作は図10のフローチャートで説明した動作と同一である。

【0058】図13は、このようにプログラムを実際に実行し、またはシミュレータなどの使用によりプロファイルした結果をもとに決定された優先順位により、早いメモリAと遅いメモリBに対しプログラムの実行速度の高速化を図れるように最適化されて構築されたメモリマップを示す。

【0059】以上のように、この実施の形態7によれば、プログラムを実際に実行し、またはシミュレータなどの使用によりプロファイルした結果をもとに決定した優先順位により、高速動作が要求される関数、変数、定数などのプログラムをアクセスの早いメモリAから遅いメモリBの順で容易に割り当て配置することができ、プログラムの実行速度の高速化を図れる最適化されたメモリマップの構築作業にかかる負荷を軽減できるメモリマップ最適化決定方法が得られる効果がある。

【0060】

【発明の効果】以上のように、この発明によれば、アクセス速度の早いメモリの順で配置すべきプログラムに対する優先順位を決定し、該決定した優先順位をもとに、前記アクセス速度の早いメモリから順に前記各プログラムを配置するに足りるメモリおよびその領域をリンカーが決定し、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを決定するように構成したので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを手間をかけずに容易に構築できる効果がある。

【0061】この発明によれば、スクリプトファイルに記述された各プログラムに対する優先順位をもとに、前記各プログラムの優先順位を決定するように構成したので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを、前記スクリプトファイルに記述された各プログラムに対する優先順位により手間をかけずに容易に構築できる効果がある。

【0062】この発明によれば、リンカー起動時にコマンドラインからオプションとして指定された優先順位をもとに、各プログラムの優先順位を決定するように構成したので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを、前記コマンドラインからオプションとして指定された優先順位により手間をかけずに容易に構築できる効果がある。

【0063】この発明によれば、各オブジェクトファイルにおいて他のオブジェクトファイルと共通的に関数毎、変数毎、定数毎に分類されたセクション毎で優先順位を決定するように構成したので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップ

を、全てのオブジェクトファイルにおいて共通的に関数毎、変数毎、定数毎に分類されたセクション毎の優先順位により手間をかけずに容易に構築できる効果がある。

【0064】この発明によれば、各オブジェクトファイルの関数毎、変数毎、定数毎に優先順位を決定するように構成したので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを、前記各オブジェクトファイルの関数毎、変数毎、定数毎の優先順位により手間をかけずに容易に構築できる効果がある。

【0065】この発明によれば、オブジェクトファイル毎に優先順位を決定するように構成したので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを、前記オブジェクトファイル毎の優先順位により手間をかけずに容易に構築できる効果がある。

【0066】この発明によれば、他のオブジェクトファイルとの間で共通的に各オブジェクトファイルにおいて関数毎、変数毎、定数毎に分類されたセクション毎と、各オブジェクトファイルの関数毎、変数毎、定数毎と、オブジェクトファイル毎の内のいずれかの組み合わせで優先順位を決定するように構成したので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを、前記セクション、前記各オブジェクトファイルの関数、変数、定数、前記オブジェクトファイルを交ぜ込んで決定した優先順位により手間をかけずに容易に構築できる効果がある。

【0067】この発明によれば、リンカーが全てのオブジェクトファイルに対して各関数、変数、定数などにアクセスしている箇所をカウントし、そのカウント値をもとに優先順位を自動的に決定するように構成したので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを、前記各関数、変数、定数などのアクセス頻度をもとに決定した優先順位により手間をかけずに容易に構築できる効果がある。

【0068】この発明によれば、プログラムを実行させてプロファイルした結果をもとに優先順位を決定するように構成したので、アクセス速度の異なる複数種類のメモリを搭載したシステム上でプログラムの実行速度の高速化を図れる最適化されたメモリマップを、前記プロファイルした結果をもとに決定した優先順位により手間をかけずに容易に構築できる効果がある。

【図面の簡単な説明】

【図1】 この発明の実施の形態1のメモリマップ最適化決定方法におけるサンプルプログラムの構成の一例を示す説明図である。

【図2】 この発明の実施の形態1のメモリマップ最適化決定方法の構成を示すフローチャートである。

【図3】 この発明の実施の形態1のメモリマップ最適化決定方法により最適化されて構築されたメモリマップ図である。

【図4】 この発明の実施の形態2のメモリマップ最適化決定方法の構成を示すフローチャートである。

【図5】 この発明の実施の形態2のメモリマップ最適化決定方法により最適化されて構築されたメモリマップ図である。

【図6】 この発明の実施の形態3のメモリマップ最適化決定方法の構成を示すフローチャートである。

【図7】 この発明の実施の形態3のメモリマップ最適化決定方法により最適化されて構築されたメモリマップ図である。

【図8】 この発明の実施の形態4のメモリマップ最適化決定方法の構成を示すフローチャートである。

【図9】 この発明の実施の形態4のメモリマップ最適化決定方法により最適化されて構築されたメモリマップ図である。

【図10】 この発明の実施の形態6のメモリマップ最適化決定方法の構成を示すフローチャートである。

【図11】 この発明の実施の形態6のメモリマップ最適化決定方法により最適化されて構築されたメモリマップ図である。

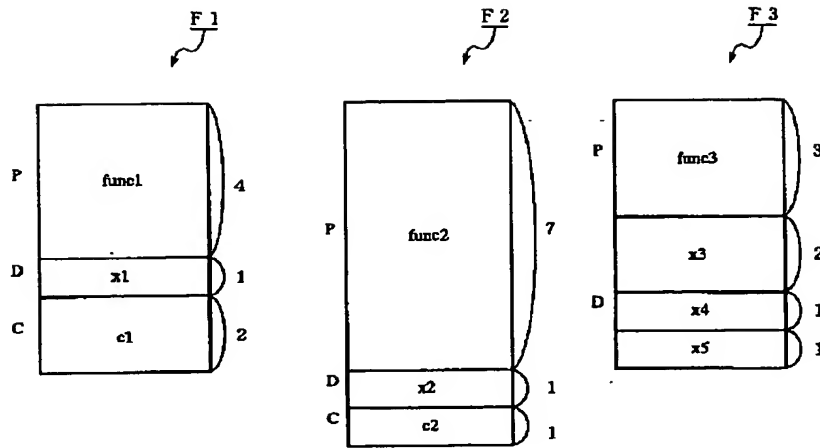
【図12】 この発明の実施の形態7のメモリマップ最適化決定方法の構成を示すフローチャートである。

【図13】 この発明の実施の形態7のメモリマップ最適化決定方法により最適化されて構築されたメモリマップ図である。

【符号の説明】

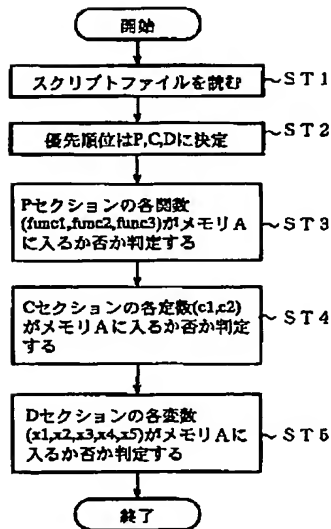
F1, F2, F3 オブジェクトファイル、ステップST1, ステップST2, ステップST11, ステップST12, ステップST21, ステップST22, ステップST31, ステップST32, ステップST41, ステップST42, ステップST51, ステップST52
優先順位決定ステップ、ステップST3～ステップST5, ステップST13～ステップST18, ステップST23～ステップST25, ステップST33～ステップST36, ステップST43～ステップST48
メモリマップ最適化ステップ。

【図1】



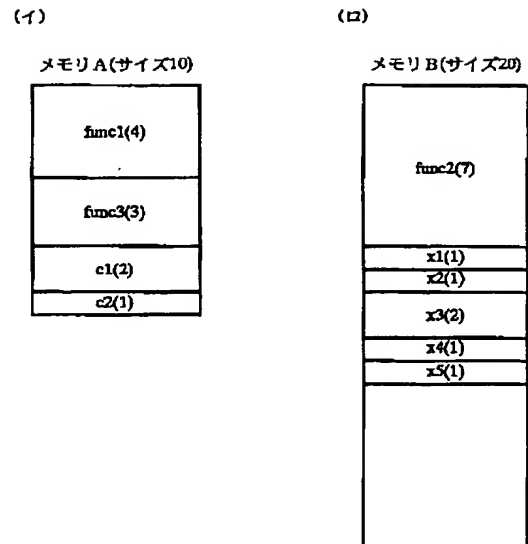
F1~F3: オブジェクトファイル

【図2】



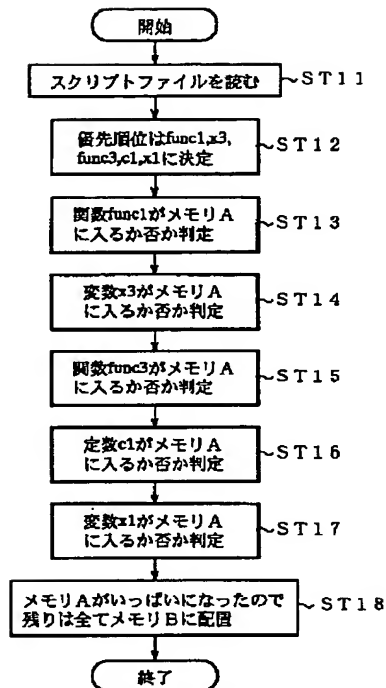
ステップST1, ステップST2: (優先順位決定ステップ)
ステップST3~ステップST5: (メモリマップ最適化ステップ)

【図3】



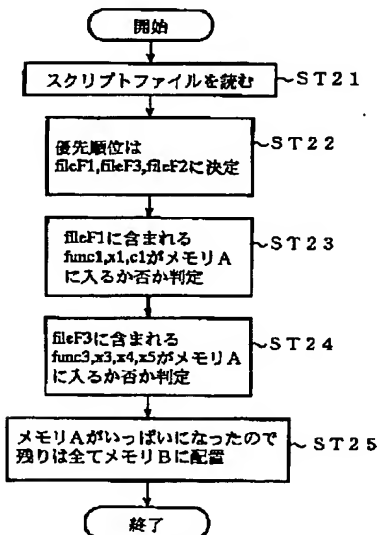
(括弧内の数字はサイズ)

【図4】



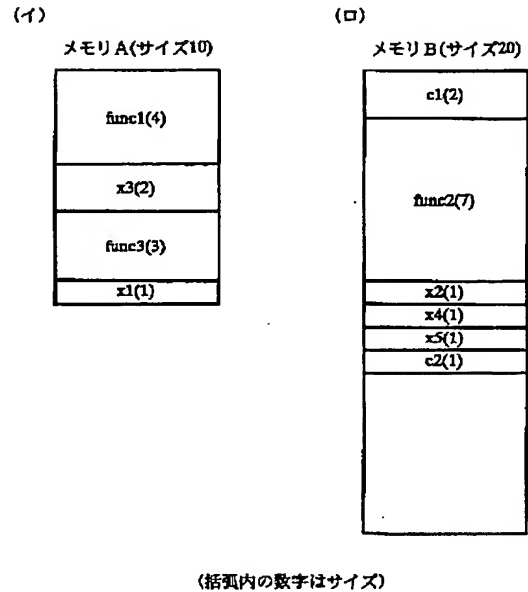
ステップST11, ステップST12: (優先順位決定ステップ)
 ステップST13~ステップST18: (メモリマップ最適化ステップ)

【図6】

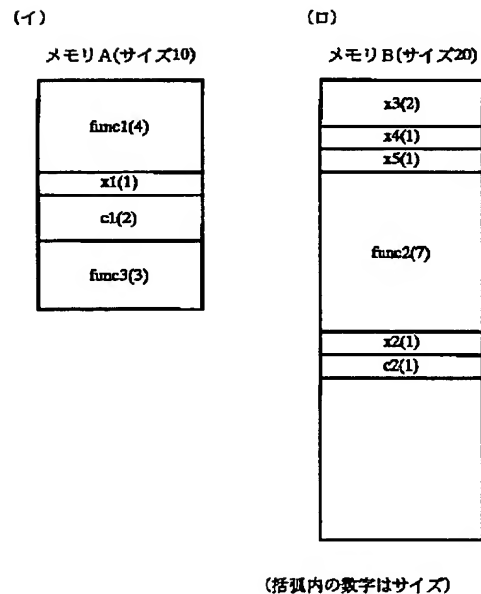


ステップST21, ステップST22: (優先順位決定ステップ)
 ステップST23~ステップST25: (メモリマップ最適化ステップ)

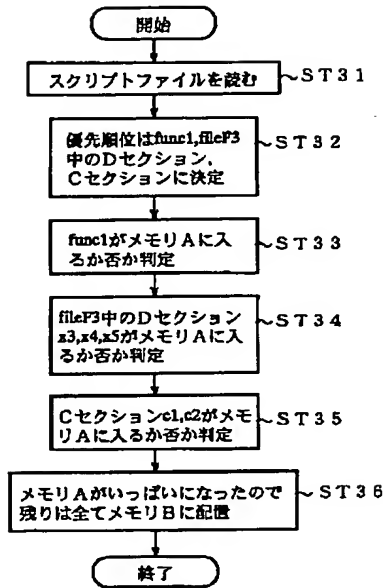
【図5】



【図7】

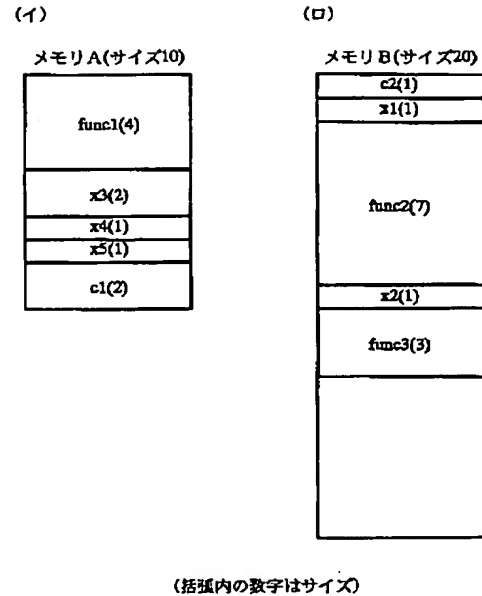


【図8】



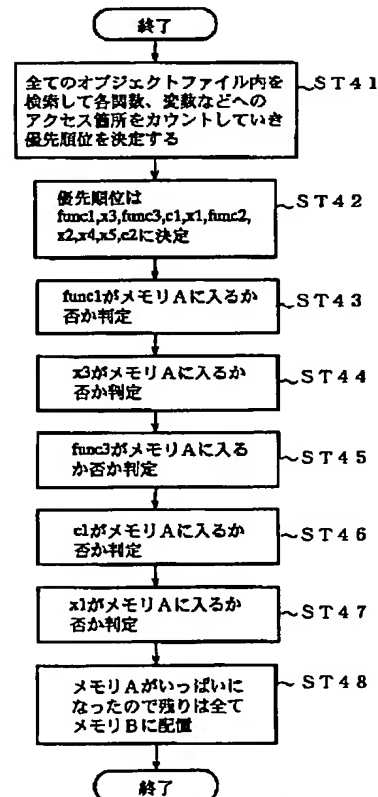
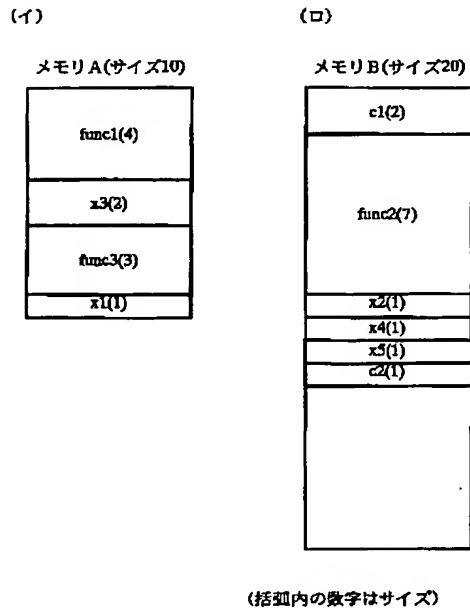
ステップST 3 1, ステップST 3 2: (優先順位決定ステップ)
 ステップST 3 3～ステップST 3 6: (メモリマップ最適化ステップ)

【図9】



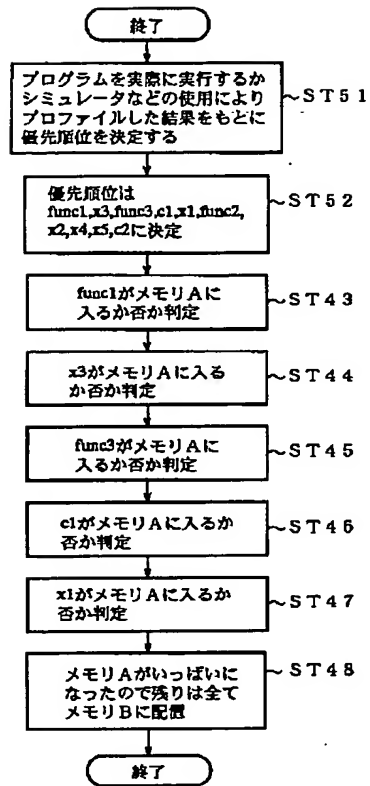
【図10】

【図11】



ステップST 4 1, ステップST 4 2: (優先順位決定ステップ)
 ステップST 4 3～ステップST 4 8: (メモリマップ最適化ステップ)

【図12】



ステップST 5 1, ステップST 5 2: (優先順位決定ステップ)

【図13】

